Programming and Problem Solving
CSIS 2610

Lab x — Planimeter

Goal

Create a program that calculates the area of a polygon.

Background

A *planimeter* is a device that calculates the area of a region by traversing the perimeter of the region. It is often used by surveyors, who can use it to calculate the area of a boundary plot by running the planimeter along the perimeter of the plot.

In this lab, we'll calculate the area of a polygon in a similar manner. By using polygons, the arithmetic involved remains simple, and a single `do-while` loop is necessary for traversing the polygon perimeter.

Details

Suppose we have a polygon consisting of vertices $p_1, p_2, \ldots, p_n$, and point p_i consists of coordinates (x_i, y_i). We can calculate the area of the polygon by using the formula

$$A = \frac{1}{2}\sum_{i=1}^{n} p_i \times p_{i+1} = \frac{1}{2}\sum_{i=1}^{n} x_i \cdot y_{i+1} - x_{i+1} \cdot y_i$$

where $p_{n+1} = p_1$.

This isn't difficult to do; the arithmetic consists of just the four basic operations. The issues involved are:

- What is the value of n?
- How do you store a point?
- How do you store a list of points?

For the first question, we don't know what n is. However, we can tell when to stop. Note that in the sequence, the last point is the same as the first point. So, if we keep track of the first point, we can tell when we reach the last point.

The second question has several possible answers, none of which are particularly difficult. For our purposes, we can store a point as two variables x and y. Note that in the program, we might need to add some designation to the x and y (*e.g.*, `xFirst` for the x-coordinate of the first point). The data type can be either integer or floating-point, depending on your needs. For this lab, use a `double` data type.

For the third question, we don't actually need to store all of the points. At any given step in the calculation, we just need three things:

- The sum of the terms so far

- The previous point p_i
- The current point p_{i+1}

You'll also need to keep track of the first point, so you can tell when the current point matches the first point.

The algorithm

This is the algorithm we came up with in class; you should start your coding process by copying it into the source code as comments.

Algorithm 1 The planimeter algorithm

```
procedure PLANIMETER
    $area \leftarrow 0$

    Get first point $(x_1, y_1)$
    Copy first point to current point $(x_{cur}, y_{cur}) \leftarrow (x_1, y_1)$

    do
        Copy current point to previous point $(x_{prev}, y_{prev}) \leftarrow (x_{cur}, y_{cur})$

        Get current point $(x_{cur}, y_{cur})$

        $area \leftarrow area + x_{prev} \cdot y_{cur} - x_{cur} \cdot y_{prev}$
    while current point does not equal first point

    $area \leftarrow area/2$
    output $area$
end procedure
```

Try it out

Add an output line to display the result and you have a program ready to be tested.

How do you know if it's right? Make up a few test cases and try it out.

Share your test cases (but not your code!) with your neighbors in the lab and see if you both get the same answers.

What to turn in

When you are satisfied that the program is correct, turn it in on blackboard.